EXHIBIT 4

Maker Docs

MakerDAO Documentation

The Maker Protocol is the platform through which anyone, anywhere can generate the Dai stablecoin against crypto collateral assets. Learn how it works.

Introduction

MakerDAO is a decentralized organization dedicated to bringing stability to the cryptocurrency economy. The Maker Protocol employs a two-token system. The first being, Dai, a collateral-backed stablecoin that offers stability. The Maker Foundation and the MakerDAO community believe that a decentralized stablecoin is required to have any business or individual realize the advantages of digital money. Second, there is MKR, a governance token that is used by stakeholders to maintain the system and manage Dai. MKR token holders are the decision-makers of the Maker Protocol, supported by the larger public community and various other external parties.

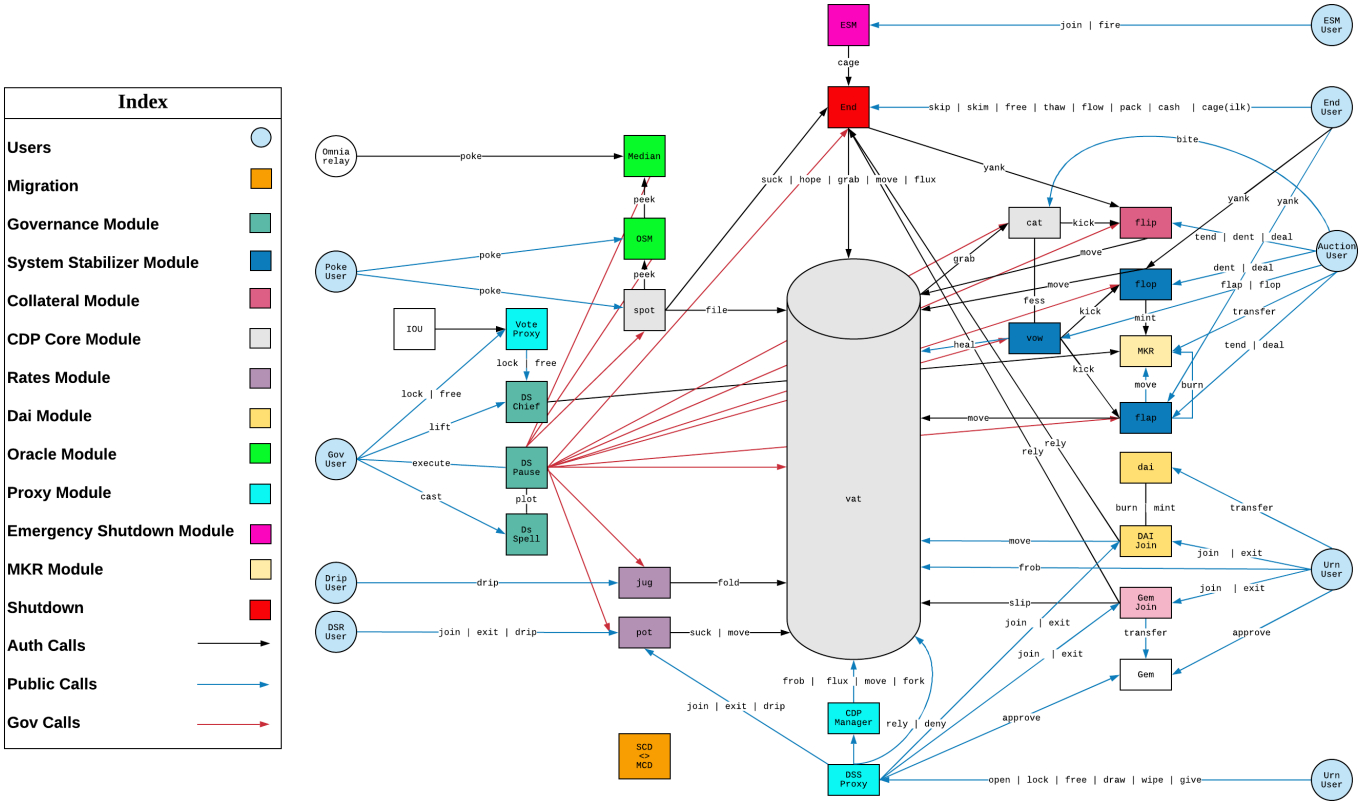
Maker is unlocking the power of decentralized finance for everyone by creating an inclusive platform for economic empowerment; enabling everyone with equal access to the global financial marketplace.

With the new version of the **Maker Protocol**, Multi Collateral Dai (MCD), being released and live on the main Ethereum network, we wanted to go over a few of the changes and features that it comes with. The biggest change to the Maker Protocol is that it now accepts any Ethereum-based asset as collateral to generate Dai given that it has been approved by MKR holders and has been given specific, corresponding Risk Parameters through the Maker decentralized governance process.

Additionally, there are a few other newly introduced features that come with the MCD upgrade. **These new features include:**

- New Dai token ($DAI)
- Support for multiple Vault collateral types (Launching with ETH and BAT)
  - To open a Vault, head to **Oasis Borrow**
- Dai Savings Rate (DSR)
  - To use the DSR, head to **Oasis Save**
- More robust peg ensuring mechanisms (MKR acting as backstop)
- Stability fees paid every block, rather than on Dai repayment
- New Maker Terminology
- MKR and governance remains the same
- Oasis Trade

The Maker Protocol Smart Contract Modules System



The Maker Protocol System Diagram

Search (Press "/" to focus)

English

No Available Languages
Need another language?
Join translation team

English

No Available Languages
Need another language? Join translation team

These are legacy guides and will not be maintened. You may be looking for the page on **Oracles**

Oracles

What is an Oracle?

An Oracle makes both off-chain and on-chain data available for use in smart-contracts. In the Maker Protocol, Oracles enable the use of price data of various assets to determine a number of important things like when to **Liquidate** a **Vault** or how much Dai a given Vault can generate. MakerDAO Oracles receive data from a number of independent Feeds that consist of individuals and organizations. Each Oracle corresponds to a single asset and its reference price.

How does the Oracle system work?

In the Maker Protocol, each collateral type has a corresponding Oracle that publishes a reference price that the system uses. Each

Each Feed uses a tool called **Setzer** ↗ which pulls the median price from a set of exchanges and then pushes it to a **Secure Scuttlebutt Network** ↗ that has relayers reading from it. Relayers aggregate the price data and send a transaction to the Medianizer. The Medianizer then takes the median of the multiple reported medians and publishes it as a queued reference price. This price is then delayed by the Oracle Security Module before it is finally used by the system.

Feeds may configure Setzer to pull from any exchanges of their choosing. Relayers are able to configure parameters around when to push price data to the Medianizer. Only MKR **governance** can configure or change the Medianizer and Oracle Security Module.

How is the Oracle system made secure?

To defend against fraudulent price-data, the reporting is decentralized; there are multiple organizations and individuals who report price-data, they are called Feeds. At the launch of Multi-Collateral Dai, Oracles received data from a total of 20 Feeds which consisted of 15 individuals and five public organizations. The Oracles use the median of the reported prices for each asset as the reference price. Using a median instead of an average makes it harder to manipulate the reference price since control over half of the data providers is needed in order for a fraudulent price to be pushed through. Additionally, using a median naturally filters out irregular price data.

In addition to this, the **Oracle Security Module(OSM)** ↗ safeguards the process by delaying price-feed data for one hour. This allows MKR governors and other stakeholders the time to identify bugs or attacks on the price-feed system. An OSM is active on all Oracles in the Maker Protocol.

Search (Press "/" to focus)

Oracle Security Modules(OSMs) ↗ delay the publishing of new reference prices for a predefined set of time. This parameter is called the `Oracle Security Module Delay` and was set to be one hour at the launch of MCD. This allows MKR token holders and other stakeholders the time to react to bugs or attacks on the Oracles. An OSM is active on each Oracle in the Maker Protocol.

Can MakerDAO governance change the time of the Oracle Security Module's delay?

Yes. This parameter is called the `Oracle Security Module Delay` and can be adjusted by MKR token holders.

What is a Medianizer?

A **Medianizer** ↗ is a type of smart-contract in the Maker Protocol's Oracle system that collects price-data from Feeds and calculates a reference price by calculating a median. The Medianizer maintains a white-list of Feeds that can be controlled by MakerDAO **governance.** Every time a new set of price updates is received, the reference price is recalculated and queued into the Oracle Security Module which publishes the price after a delay period.

How often does the Medianizer publish an updated reference price?

What is a Secure Scuttlebutt Network?

Secure Scuttlebutt is a database protocol for unforgeable append-only message feeds. For more information on how a Secure Scuttlebutt Network works visit this **informative page** ↗ on **scuttlebot.io** ↗.

Why are Oracles an attack target for malicious actors?

In the Maker Protocol, if the reference price for an asset was determined by a single party, then they could fraudulently report an incorrect price and cause severe issues. For instance, if the price of ETH was reported to be fraudulently low, say $0.01, then every single ETH **Vault** in the system would be **Liquidated**. On the other hand, if the price of ETH was reported to be artificially high, say $1,000,000.00, then any ETH Vault owner would be able to issue an excessive amount of Dai since the system thinks there is more Collateral value than there actually is. Oracle attacks can be very profitable for a successful attacker and can be very disruptive to MakerDAO and its users.

Who are the Feeds?

The Feeds are a mix of pseudonymous individuals and public organizations. Individuals consist of people internal to Maker, influential people in the greater crypto community, as well as some community members. The organizations involved in being Feeds at the launch of MCD can be found in the ratified **DeFi Feeds proposal** ↗.

pseudonymous?

From their onset, the individuals running Feeds have been pseudonymous out of necessity, to protect the individuals from the risk of extortion and blackmail. Pseudonymous Feeds also have the benefit of making it harder to coordinate an Oracle attack since the Feeds don't know who the others are. Organizations running Feeds, however, are different. Organizations are much more resilient against coercion, have the resources to combat malicious actors, and have their reputations at stake. This makes them much better equipped to be Feeds with public identities. A hybrid model is optimal, one that incorporates the benefits of both individual and organizational Feeds.

What is the process for becoming a Feed?

All new Feeds go through MakerDAO's **governance** in order to be added in. There is currently no formal way for Feeds to be added to the Maker Protocol. As of October 07th, 2019, the ratified **Oracle Team Mandate** ↗ grants the Interim Oracle Team the responsibility of being the intermediary between the Feeds and governance. In the coming months, the process of becoming a Feed will become more clear.

Is the Oracle system resistant to Sybil attacks?

To quote from Wikipedia's page on **Sybil Attacks** ↗, "In a Sybil attack, the attacker subverts the reputation system of a peer-to-peer network by creating a large number of pseudonymous identities, using them to gain a disproportionately large influence. A reputation system's vulnerability to a Sybil attack depends on how cheaply identities can be generated, the degree to which the

The short answer is yes, the Oracle system is resistant to Sybil attacks because of the existence of a whitelist for Feeds. It's not simple to become a Feed, they need to be approved by MKR **governance.** Therefore, an attacker cannot gain a majority influence by creating many pseudo-feeds.

What happens if there is a flash crash on an exchange?

Since the reference price published by the Oracles is a median of the median prices that are reported by at least 20 different Feeds, outliers are automatically filtered out. In practice, this means if a single exchange experiences a flash crash the set of prices will look something like this:

```
[0.70, 104.80, 104.82, **104.88**, 104.90, 105.02, 10
```

The median of this set still reflects the real market price of the asset. Flash crashes on single exchanges do not affect the published reference price.

Where can I find more technical information about Oracles?

Visit our **Documentation Portal** ↗ for all technical documentation of the Maker Protocol. Technical documentation of Oracles can be found in the **Oracle Module** ↗ section of our Documentation Portal.

Search (Press "/" to focus)

Resources

Whitepaper

FAQs

Privacy Policy

Brand Assets

Feeds

Service Status

Products

Oasis

Migrate

Ecosystem

Governance

Developers

Documentation

Dai.js

Developer Guides

Foundation

Contact

Blog